A Comparative Study of Web Services-based Event Notification Specifications

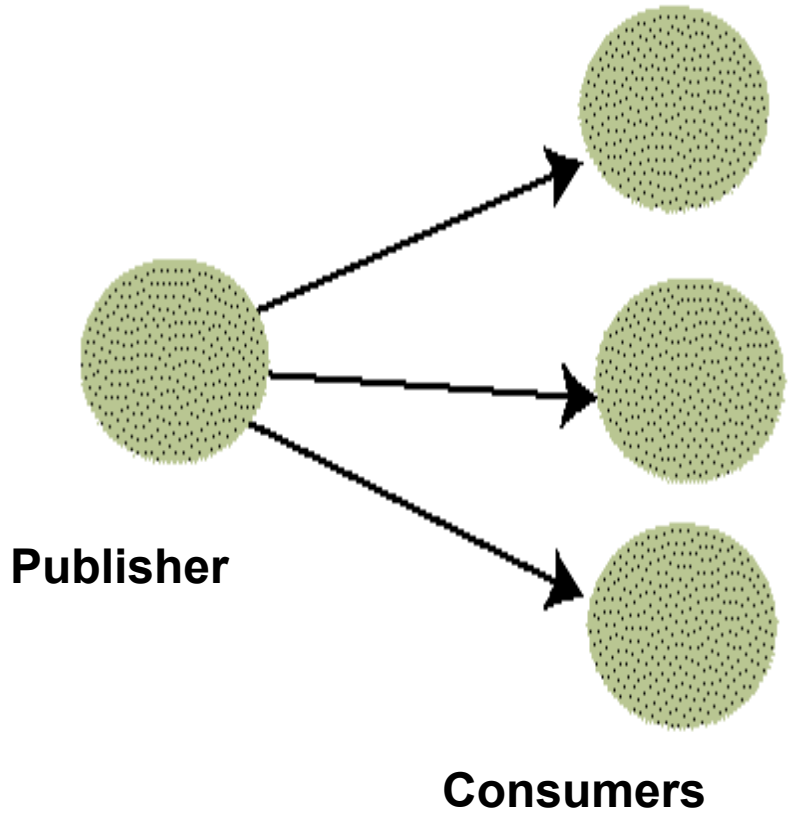
Yi Huang and Dennis Gannon
Extreme! Computing Lab
Dept. of Computer Science
Indiana University

Event Notification Systems

- Enable information disseminations, e.g news, stock quotes
- Event notification is an important component in distributed computing
  - Logging, auditing, monitoring
- Enable asynchronous communications
  - Results of long running processes
- Event-driven architecture (EDA)
  - Web services interact with each other by events
  - Decouple services

Publish/Subscribe Model

- an **event consumer** registers its interest for some specific kind of events using a "***subscribe***" operation
- an **event source** "***publishes***" events to one or more event consumers based on their registered interests.



Notification Brokers

- provides an abstraction layer between an event source and an event consumer
- Loosely-coupled communication model
- communicate without knowing the location of each other.
  - The event consumer can be offline when the event source publishes an event.
  - The event source is relieved from the burden of handling subscription registrations and delivering events to all the event consumers.

Web services-based Publish/Subscribe systems

- Use web services technologies to provide **interoperability** for event notification systems
- **Web services** can provide
  - *Programming Language-, Platform-, Transport-* Independence
  - Great fit for integrating *heterogeneous* resources

Evolution of Event Notification specifications

- CORBA *(Common Object Request Broker Architecture)* Event Service (3/1995)
- CORBA Notification Service (6/1997)
- Java Message Service (JMS) (4/2002)
- OGSI-Notification (6/2003)
  - Open Grid Services Infrastructure
- WS-Eventing (1/7/2004)
- WS-Notification (1/20/2004)

Outline

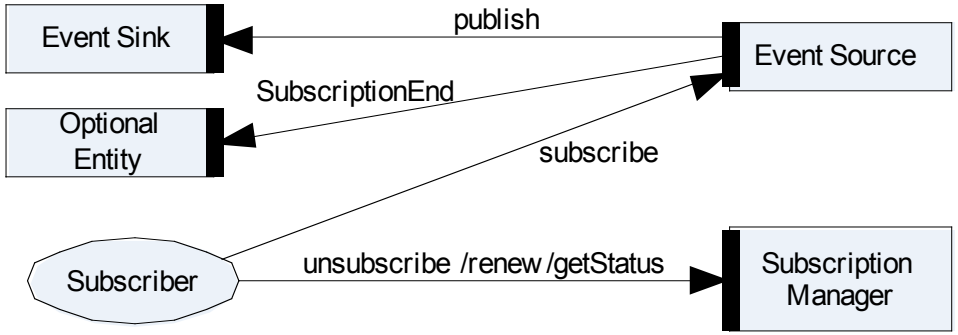
- Introduction
- Compare WS-Eventing and WS-Notification
- Compare WS-based Pub/Sub specifications with previous specifications
- WS-Messenger project

Competing Specifications

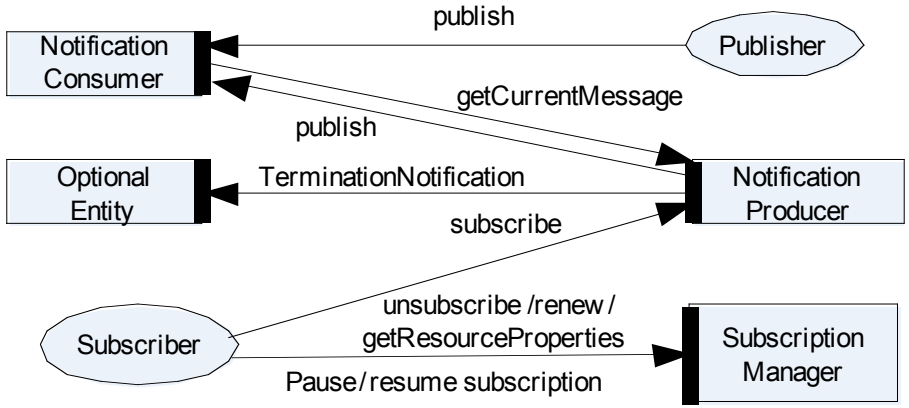
- **WS-Notification**
  - IBM, Sonic, TIBCO, Akamai, SAP, CA, HP Fujitsu, Globus
  - Implemented in **Globus Toolkit 4 (GT4), Apache ServiceMix, Apache Pubscribe, IBM ETTK toolkit, WS-Messenger**
  - Currently in OASIS. Version 1.3 published as "Committee Specification" on July 31,2006
  - WS-ResourceFramework (WSRF) is optional in Version 1.3
  - WS-BaseNotification, WS-BrokeredNotification, WS-Topic
- **WS-Eventing**
  - IBM, BEA, CA, Sun, Microsoft, TIBCO
  - Simpler and easier to use
  - Implemented in **Apache Pubscribe, FINS (NaradaBrokering), "Plumbwork Orange"(MS .net), WS-Messenger**
  - latest version is the 8/2004 version
  - Proposed as alternative implementation for Open Grid Service Architecture (OGSA) (U. Of Virginia)
- **Incompatible with each other**

Architecture Comparison

WS-Eventing



WS-Notification



Function Comparison

WS-Eventing	WS-BaseNotification
Subscribe	Subscribe
Renew	Renew
Unsubscribe	Unsubscribe
GetStatus	Not defined, can use getResourceProperties in WSRF
SubscriptionEnd	Not defined, can use TerminationNotification in WSRF
Not available	Pause/resume Subscription (optional)
Not available	GetCurrentMessage (optional)

XML Message Formats Comparison

- Element names or attribute names difference
- Namespaces difference
- Versions difference of underlying specifications
- Message contents difference
- SOAP message structures difference
- Content locations difference

WSE and WSN are Converging…

	WSE 01/04	WSN 1.0	WSE 08/04	WSN 1.3
Version date	**1/2004**	**3/2004**	**8/2004**	**2/2006**
Separate Subscription Manager & Event Source	No	Yes	Yes	Yes
Separate subscriber & Event Sink	No	Yes	Yes	Yes
Getstatus operation	No	Yes	Yes	Yes
Return subscriptionId in WSA of Subscription Manager	No	Yes	Yes	Yes
Support Wrapped delivery mode	No	Yes	Yes	Yes
Support Pull delivery mode	No	No	Yes	Yes
Specify subscription expiration using duration	Yes	No	Yes	Yes
Specify XPath dialect	Yes	No	Yes	Yes
Filter element in Subscription message	Yes	No	Yes	Yes
Require WSRF	No	Yes	No	No
Require a topic in subscription	No	Yes	No	No
Require Pause/Resume subscriptions	No	Yes	No	No

Still have differences

	WSE 01/04	WSN 1.0	WSE 08/04	WSN 1.3
Version date	1/2004	3/2004	8/2004	2/2006
GetCurrentMessage operation	No	Yes	No	Yes
Define Wrapped message format	No	Yes	No	Yes
Separate EventProducer & Publisher	No	Yes	No	Yes
Define PullPoint interface	No	No	No	Yes
Specify pull delivery mode in subscription	No	No	Yes	No
Require Getstatus	Yes	Yes	Yes	No
Require SubscriptionEnd	Yes	Yes	Yes	No
WS-Addressing version	2003/03	2003/03	2004/08	2005/08

Outline

- Introduction
- Compare WS-Eventing and WS-Notification
- Compare WS-based Pub/Sub specifications with previous specifications
- WS-Messenger project

Previous Specifications

- CORBA Event services (1995)
  - Common Object Request Broker Architecture
  - Not event filtering & QoS in Event services
- CORBA Notification services (1997)
- Java Message Service (JMS) (1998)
- OGSI notification (2003)
  - Open Grid Services Infrastructure
  - Replaced by WS-Notification

Overview

	CORBA Event Service	CORBA Notification Service	JMS	OGSI-Notification	WS-Notification	WS-Eventing
First Release	3/95	6/97	1998	6/27/03	1/20/04	1/7/04
Latest Release	10/2/04	10/11/04	4/12/02	6/27/03	7/31/06	8/30/04
Creator(s)	OMG	OMG	Sun	Global Grid Forum	IBM, Sonic, TIBCO, Akamai, SAP, CA, HP Fujitsu, Globus	IBM, BEA, CA, Sun, Microsoft, TIBCO

Delivery

	CORBA Event Service	CORBA Notification Service	JMS	OGSI-Notification	WS-Notification	WS-Eventing
Message transport	RPC	RPC	RPC	HTTP RPC	Transport independent	Transport independent
Intermediary	Event Channel object	Event Channel object	Message Queue, Pub/Sub broker	directly or through interme diary	directly or through broker	directly or through broker
Delivery Mode	Push, pull & both	Push, pull & both	Pull, Push	Push	Push, Pull	Push by default, Can use Pull or other modes

Message

	CORBA Event Service	CORBA Notification Service	JMS	OGSI-Notification	WS-Notification	WS-Eventing
Message Structure	Generic (Anys), Typed	Generic (Anys), Typed, Structured, sequences of structured	TextMessage, ByteMessage, MapMessage, StreamMessage, ObjectMessage	Xml based Service data Elements	SOAP (with Raw XML data or wrapped messages)	SOAP (with Raw XML data only). Can use wrapped mode.
Filter	No	Channel, Filter Object.	Queue/topic name, message selector on header fields	ServiceDataName. Can add other filter services.	Hierarchy Topic tree; Content Selector. Producer properties.	A "Filter" element for any filter. At most 1 filter.
Filter language	No	Extended Trader Constraint Language	a subset of the SQL92 conditional expression syntax	ServicedDataName String or other expressions.	Any expression (xsd:any) that evaluates to a Boolean. e.g. XPath	Default XPath. Can use any expression (xsd:any) that evaluates to a Boolean.

Management

	CORBA Event Service	CORBA Notification Service	JMS	OGSI-Notification	WS-Notification	WS-Eventing
QoS criteria	Not defined	Defined 13 QoS properties, can be extended to others	Priority; persistence; durable; transaction; message order	Not defined	Depends on composition with other WS* specification	Depends on composition with other WS* specification
Subscription Timeout	No	No	No	Absolute Time	Absolute Time or duration	Absolute time or duration
Demand-based	No	Defined	No	No	Defined	No
Management operations	connect_*, obtain _(typed)_push/pull_supplier/consumer	connect_*, obtain_notification_pull/push_supplier/consumer, suspend/resume_connection., get/set/validate_qos, add/remove/get/getAll/removeAll_filter, obtain_subscription/offered_types	createSubscriber, createDurableSubscriber, unsubscribe	Subscribe, requestTerminationAfter, requestTerminationBefore, destroy	Subscribe,Renew, unsubscribe, Pause/resume subscription, get/getMultiple/set/query ResourceProperties, TerminationNotification, Destroy, SetTerminationTime	Subscribe, Renew, GetStatus, Unsubscribe, SubscriptionEnd

Major Changes in WS Pub/Sub

- The event delivery scope is extended to the Internet scale.
  - delivery mechanism is moving towards *transport-independent*.
- Payloads: XML-based SOAP messages.
- The message filtering mechanism
  - simple *subject-based* topic filtering =>content-based XPath filtering.
- The criteria of Quality of Service (QoS) are no longer defined in the specifications.
  - e.g. reliability, transaction depend on the composition with other WS-* specifications, such as WS-Reliability, WS-Transaction.

Major Changes in WS Pub/Sub

- The soft-state management (timeout) of subscription terminations is used.
  - The connections to event consumers do not always keep alive.
- Interoperability concerns are shifted from the fine-grained API level to the more coarse-grained service interfaces and SOAP messages level.
  - Event producers, event consumers and brokers can interoperate with each other using SOAP messages with standard formats.
  - They do not need to use implementations from the same vendor

Outline

- Introduction
- Compare WS-Eventing and WS-Notification
- Compare WS-based Pub/Sub specifications with previous specifications
- WS-Messenger project

WS-Messenger project

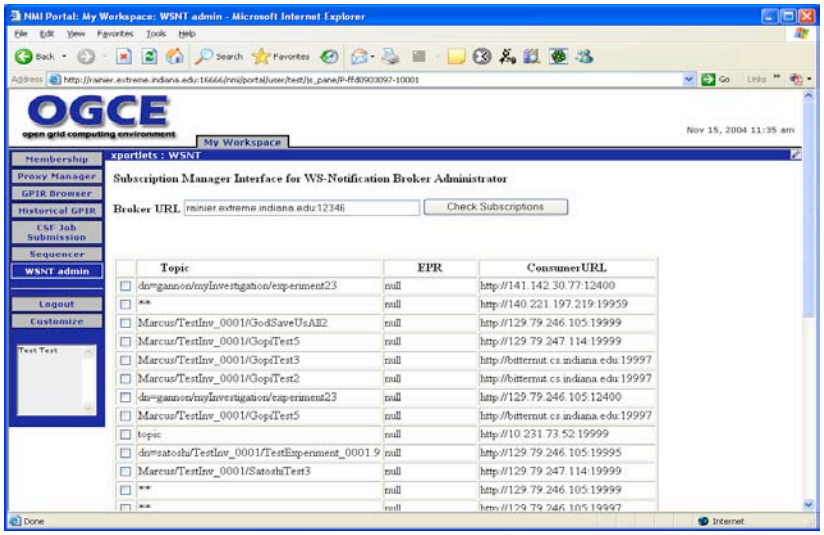
- designed to create a ***scalable***, ***reliable*** and ***efficient*** Web services-based message broker that
  - sends Web services based event notification messages
  - among heterogeneous applications, platforms and Grid computing environments
- The research efforts for WS-Messenger focus on addressing ***unique challenges*** in Web services based publish/subscribe systems.

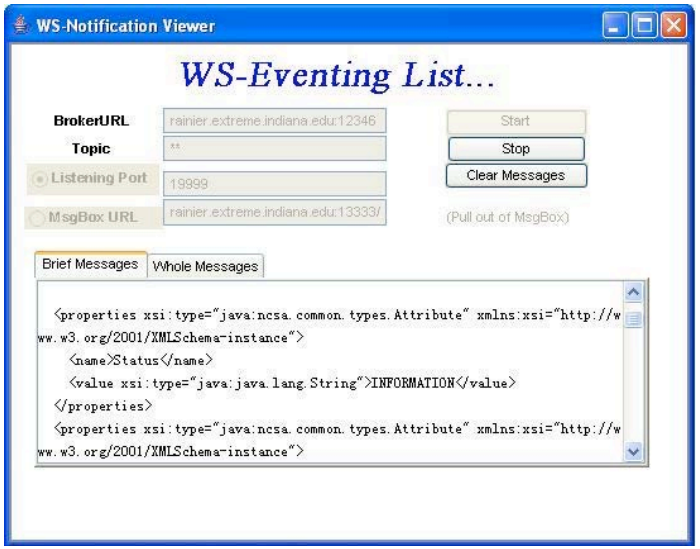
Key Features

- Based on Web services specifications and provides mediation between WS-Notification specification and WS-Eventing specification.
- provides an extensible framework to leverage different existing underlying messaging systems so that it can adapt to different environments.
- light-weighted and has simple-to-use APIs to integrate with existing Java applications.
- provides graphic interfaces for subscription management and debugging Web services based publish/subscribe systems.

Supporting Tools

- Subscription Manager Interface
  - Check and delete subscriptions on different brokers from one simple interface
- Notification Message Viewer
  - Debug
  - Monitor
  - Firewall

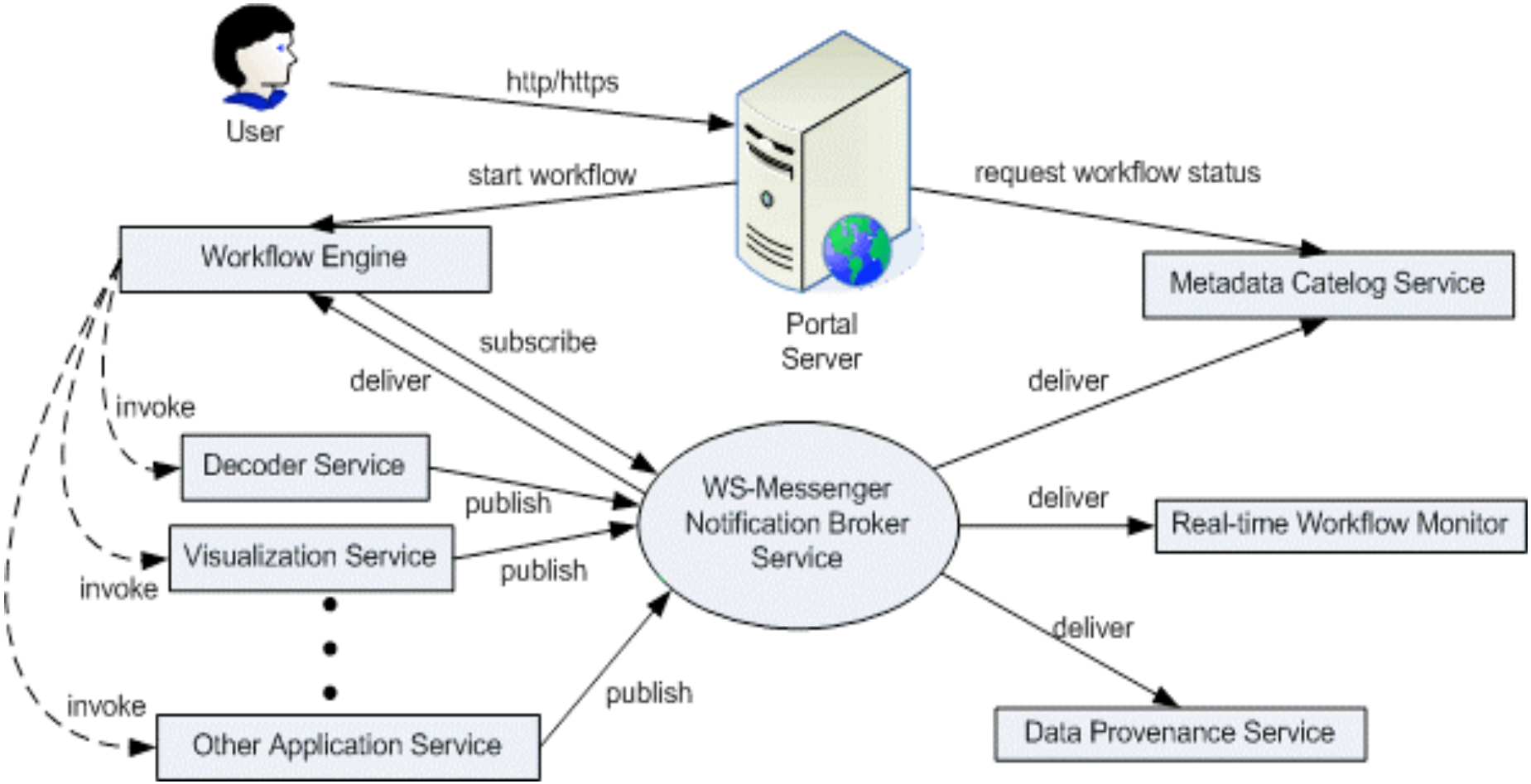




LEAD project

- **L**inked **E**nvironments for **A**tmospheric **D**iscovery (LEAD) project
- A Grid project that
  - addresses the limitations of current weather forecast frameworks
  - through a new, service-oriented architecture
  - capable of responding to unpredicted weather events and response patterns in real time

Application in the LEAD project (Linked Environments for Atmospheric Discovery)



Conclusions

- Compared WS-Eventing and WS-Notification
  - Take ideas and concepts from each other
  - Have a trend of convergence
- Studied the evolution of event notification specifications
  - Identified key shifts in WS Pub/Sub systems
- WS-Messenger addresses unique challenges in WS-Pub/Sub systems
  - Provide mediation between WSE and WSN
  - Working on Internet scale XPath-based filtering and scalability

Thank you!

- More information and download WS-Messenger
  - http://extreme.indiana.edu/xgws/messenger.html
  - or google “WS-Messenger”
- LEAD project: http://www.leadproject.org
- Email: yihuan@cs.indiana.edu